Laboratory 3: Postlab

Date 21/09/2022 Section

Name Anudeep Gadige

Step 1

Create a program called *lab3.4.asm* as follows:

- reserve space in memory for an array of words of size 10. Use the '.space' directive. The array is called *my_array*.
- the program will implement the piece of C code described below. The value of `initial_value` is the first digit of your SSN. `i` and `j` will be in one of the registers **`$t0`** to **`$t9`**.

```
j = initial_value;
for (i=0; i<10, i++) {
      my_array[i] = j;
      j++;
}
```

Run the program and make sure it works. Do not forget the comments at each line of code indicating what they do.

Hint: A common mistake here is to forget that sequential word addresses in memory differ by 4 not by one.

Step 2

Most branches have as a target an instruction that is nearby. Occasionally however, a branch may have a target that is very far away, much farther than can be represented using the 16 bit offset. Write a program called *lab3.5.asm* that shows such a situation. The description of the program follows:

- prompts the user to enter two integers; store them in **`$t0`** and **`$t1`**
- if the two integers are equal, then the program branches to a label called 'Far' that is very far away (farther than a 16 bit offset can indicate), prints the message "I'm far away" and terminates.
- if the two integers are different, then the program prints the message "I'm nearby" and terminates.

Q 1:

What is the sequence of instructions the assembler generates to implement this branch?

Synthetic Instruction	Native Instructions	Effect
bne $t0, $t1, close	bne $8, $9, 24 [close-0x00400058]	if($t0!=$t1) PC <- close
j exit	j 0x00400080 [exit]	PC <- j label
jr $ra	jr $31	PC <- $ra

Step 3

Return to your lab instructor copies of *lab3.4.asm* and *lab3.5.asm* together with this postlab description. Ask your lab instructor whether copies of programs must be on paper (hardcopy), e-mail or both.